Introduction to RBM package

Dongmei Li

May 9, 2023

Clinical and Translational Science Institute, University of Rochester School of Medicine and Dentistry, Rochester, NY 14642-0708

Contents

1 Overview

This document provides an introduction to the `RBM` package. The `RBM` package executes the resampling-based empirical Bayes approach using either permutation or bootstrap tests based on moderated t-statistics through the following steps.

- Firstly, the RBM package computes the moderated t-statistics based on the observed data set for each feature using the lmFit and eBayes function.
- Secondly, the original data are permuted or bootstrapped in a way that matches the null hypothesis to generate permuted or bootstrapped resamples, and the reference distribution is constructed using the resampled moderated t-statistics calculated from permutation or bootstrap resamples.
- Finally, the p-values from permutation or bootstrap tests are calculated based on the proportion of the permuted or bootstrapped moderated t-statistics that are as extreme as, or more extreme than, the observed moderated t-statistics.

Additional detailed information regarding resampling-based empirical Bayes approach can be found elsewhere (Li et al., 2013).

2 Getting started

The RBM package can be installed and loaded through the following R code.
Install the RBM package with:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("RBM")
```

Load the RBM package with:

```
> library(RBM)
```

3 RBM_T and RBM_F functions

There are two functions in the RBM package: RBM_T and RBM_F. Both functions require input data in the matrix format with rows denoting features and columns denoting samples. RBM_T is used for two-group comparisons such as study designs with a treatment group and a control group. RBM_F can be used for more complex study designs such as more than two groups or time-course studies. Both functions need a vector for group notation, i.e., "1" denotes the treatment group and "0" denotes the control group. For the RBM_F function, a contrast vector need to be provided by users to perform pairwise comparisons between groups. For example, if the design has three groups (0, 1, 2), the aContrast parameter will be a vector such as ("X1-X0", "X2-X1", "X2-X0") to denote all pairwise comparisons. Users just need to add an extra "X" before the group labels to do the contrasts.

- Examples using the RBM_T function: normdata simulates a standardized gene expression data and unifdata simulates a methylation microarray data. The p-values from the RBM_T function could be further adjusted using the p.adjust function in the stats package through the Bejamini-Hochberg method.

```
> library(RBM)
> normdata <- matrix(rnorm(1000*6, 0, 1),1000,6)
> mydesign <- c(0,0,0,1,1,1)
> myresult <- RBM_T(normdata,mydesign,100,0.05)
> summary(myresult)

              Length Class  Mode
ordfit_t      1000   -none- numeric
ordfit_pvalue 1000   -none- numeric
ordfit_beta0  1000   -none- numeric
ordfit_beta1  1000   -none- numeric
permutation_p 1000   -none- numeric
bootstrap_p   1000   -none- numeric

> sum(myresult$permutation_p<=0.05)
```

```
[1] 25

> which(myresult$permutation_p<=0.05)

 [1]   1   6  27  53  82 105 131 163 185 281 326 348 358 378 554 564 566 611 734
[20] 793 801 821 893 960 963

> sum(myresult$bootstrap_p<=0.05)

[1] 0

> which(myresult$bootstrap_p<=0.05)

integer(0)

> permutation_adjp <- p.adjust(myresult$permutation_p, "BH")
> sum(permutation_adjp<=0.05)

[1] 2

> bootstrap_adjp <- p.adjust(myresult$bootstrap_p, "BH")
> sum(bootstrap_adjp<=0.05)

[1] 0

> unifdata <- matrix(runif(1000*7,0.10, 0.95), 1000, 7)
> mydesign2 <- c(0,0,0, 1,1,1,1)
> myresult2 <- RBM_T(unifdata,mydesign2,100,0.05)
> sum(myresult2$permutatioin_p<=0.05)

[1] 0

> sum(myresult2$bootstrap_p<=0.05)

[1] 4

> which(myresult2$bootstrap_p<=0.05)

[1] 288 345 734 829

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=0.05)

[1] 0
```

- Examples using the `RBM_F` function: normdata_F simulates a standardized gene expression data and unifdata_F simulates a methylation microarray data. In both examples, we were interested in pairwise comparisons.

```
> normdata_F <- matrix(rnorm(1000*9,0,2), 1000, 9)
> mydesign_F <- c(0, 0, 0, 1, 1, 1, 2, 2, 2)
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult_F <- RBM_F(normdata_F, mydesign_F, aContrast, 100, 0.05)
> summary(myresult_F)

              Length Class  Mode
ordfit_t      3000   -none- numeric
ordfit_pvalue 3000   -none- numeric
ordfit_beta1  3000   -none- numeric
permutation_p 3000   -none- numeric
bootstrap_p   3000   -none- numeric

> sum(myresult_F$permutation_p[, 1]<=0.05)

[1] 72

> sum(myresult_F$permutation_p[, 2]<=0.05)

[1] 76

> sum(myresult_F$permutation_p[, 3]<=0.05)

[1] 85

> which(myresult_F$permutation_p[, 1]<=0.05)

 [1]  11  31  65  75  93 111 114 117 128 142 166 171 186 198 201 202 221 226 243
[20] 258 259 270 276 282 287 288 314 326 366 387 406 407 432 446 481 492 493 505
[39] 507 529 530 535 554 603 620 623 628 640 651 658 671 684 704 718 739 765 769
[58] 800 833 839 840 877 895 899 928 932 935 936 937 958 990 998

> which(myresult_F$permutation_p[, 2]<=0.05)

 [1]  11  31  65  75 111 117 128 142 154 166 171 186 198 201 202 221 226 236 243
[20] 258 259 270 275 282 287 288 306 314 328 387 406 407 432 446 463 481 492 493
[39] 505 507 529 530 535 554 603 612 620 628 640 651 658 671 684 704 718 739 765
[58] 769 778 800 802 833 839 840 846 877 895 899 932 935 936 937 958 990 992 998

> which(myresult_F$permutation_p[, 3]<=0.05)

 [1]  11  31  34  65  67  75  93 114 117 128 131 166 171 186 198 201 202 221 226
[20] 229 236 243 258 259 270 282 287 306 314 326 328 340 366 380 387 406 407 432
[39] 435 446 462 481 490 492 493 505 507 529 530 535 554 577 603 620 628 640 651
[58] 658 671 680 684 704 718 732 739 765 769 778 802 833 840 846 850 877 880 895
[77] 928 932 935 936 937 958 990 994 998
```

```
> con1_adjp <- p.adjust(myresult_F$permutation_p[, 1], "BH")
> sum(con1_adjp<=0.05/3)

[1] 13

> con2_adjp <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adjp<=0.05/3)

[1] 11

> con3_adjp <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adjp<=0.05/3)

[1] 20

> which(con2_adjp<=0.05/3)

 [1]  75 186 259 407 492 507 628 658 739 840 935

> which(con3_adjp<=0.05/3)

 [1]  65  75 117 166 186 202 282 435 492 529 603 620 628 640 651 684 718 739 840
[20] 990

> unifdata_F <- matrix(runif(1000*18, 0.15, 0.98), 1000, 18)
> mydesign2_F <- c(rep(0, 6), rep(1, 6), rep(2, 6))
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult2_F <- RBM_F(unifdata_F, mydesign2_F, aContrast, 100, 0.05)
> summary(myresult2_F)

             Length Class  Mode
ordfit_t     3000   -none- numeric
ordfit_pvalue 3000   -none- numeric
ordfit_beta1  3000   -none- numeric
permutation_p 3000   -none- numeric
bootstrap_p   3000   -none- numeric

> sum(myresult2_F$bootstrap_p[, 1]<=0.05)

[1] 53

> sum(myresult2_F$bootstrap_p[, 2]<=0.05)

[1] 52

> sum(myresult2_F$bootstrap_p[, 3]<=0.05)

[1] 62
```

```
> which(myresult2_F$bootstrap_p[, 1]<=0.05)

 [1]  74 102 117 164 202 203 204 207 257 267 286 306 315 367 373 392 408 425 432
[20] 451 464 469 481 489 495 496 505 506 535 538 551 575 576 596 597 637 655 661
[39] 681 732 765 774 790 813 841 848 851 859 890 920 953 954 978

> which(myresult2_F$bootstrap_p[, 2]<=0.05)

 [1]  14  20  65  74  95 102 117 164 176 179 202 203 204 207 267 286 306 367 373
[20] 392 432 451 464 469 481 495 496 505 506 535 538 551 575 576 596 639 644 655
[39] 656 661 681 702 733 765 774 790 848 863 890 920 953 954

> which(myresult2_F$bootstrap_p[, 3]<=0.05)

 [1]   2  14  65  74 114 117 122 164 191 203 204 207 250 267 269 286 306 367 373
[20] 376 392 403 408 432 451 464 469 481 495 496 506 521 532 535 538 551 590 596
[39] 637 639 644 655 656 661 681 732 733 765 790 808 813 841 848 859 863 890 920
[58] 921 953 954 978 991

> con21_adjp <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adjp<=0.05/3)

[1] 8

> con22_adjp <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adjp<=0.05/3)

[1] 9

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=0.05/3)

[1] 12
```

4 Ovarian cancer methylation example using the RBM_T function

Two-group comparisons are the most common contrast in biological and biomedical field. The ovarian cancer methylation example is used to illustrate the application of RBM_T in identifying differentially methylated loci. The ovarian cancer methylation example is taken from the gemone-wide DNA methylation profiling of United Kingdom Ovarian Cancer Population Study (UKOPS). This study used Illumina Infinium 27k Human DNA methylation Beadchip v1.2 to obtain DNA methylation profiles on over 27,000 CpGs in whole blood cells from 266 ovarian cancer women and 274 age-matched healthy controls. The data are downloaded from the NCBI GEO website with access number GSE19711. For illutration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovariance cancer cases (pre-treatment) and 4 healthy controls. The following codes show the process of generating significant differential DNA methylation loci using the RBM_T function and presenting the results for further validation and investigations.

```
> system.file("data", package = "RBM")

[1] "/private/tmp/RtmpSytI4B/Rinst182d45eba8604/RBM/data"

> data(ovarian_cancer_methylation)
> summary(ovarian_cancer_methylation)

        IlmnID         Beta          exmdata2[, 2]     exmdata3[, 2]
 cg00000292:  1   Min.   :0.01058   Min.   :0.01187   Min.   :0.009103
 cg00002426:  1   1st Qu.:0.04111   1st Qu.:0.04407   1st Qu.:0.041543
 cg00003994:  1   Median :0.08284   Median :0.09531   Median :0.087042
 cg00005847:  1   Mean   :0.27397   Mean   :0.28872   Mean   :0.283729
 cg00006414:  1   3rd Qu.:0.52135   3rd Qu.:0.59032   3rd Qu.:0.558575
 cg00007981:  1   Max.   :0.97069   Max.   :0.96937   Max.   :0.970155
 (Other)   :994                     NA's   :4
 exmdata4[, 2]     exmdata5[, 2]     exmdata6[, 2]     exmdata7[, 2]
 Min.   :0.01019   Min.   :0.01108   Min.   :0.01937   Min.   :0.01278
 1st Qu.:0.04092   1st Qu.:0.04059   1st Qu.:0.05060   1st Qu.:0.04260
 Median :0.09042   Median :0.08527   Median :0.09502   Median :0.09362
 Mean   :0.28508   Mean   :0.28482   Mean   :0.27348   Mean   :0.27563
 3rd Qu.:0.57502   3rd Qu.:0.57300   3rd Qu.:0.52099   3rd Qu.:0.52240
 Max.   :0.96658   Max.   :0.97516   Max.   :0.96681   Max.   :0.95974
                   NA's   :1
 exmdata8[, 2]
 Min.   :0.01357
 1st Qu.:0.04387
 Median :0.09282
 Mean   :0.28679
 3rd Qu.:0.57217
 Max.   :0.96268

> ovarian_cancer_data <- ovarian_cancer_methylation[, -1]
> label <- c(1, 1, 0, 0, 1, 1, 0, 0)
> diff_results <- RBM_T(aData=ovarian_cancer_data, vec_trt=label, repetition=100, alpha=0.05)
> summary(diff_results)

             Length Class  Mode
ordfit_t     1000   -none- numeric
ordfit_pvalue 1000   -none- numeric
ordfit_beta0  1000   -none- numeric
ordfit_beta1  1000   -none- numeric
permutation_p 1000   -none- numeric
bootstrap_p   1000   -none- numeric

> sum(diff_results$ordfit_pvalue<=0.05)

[1] 45
```

```
> sum(diff_results$permutation_p<=0.05)

[1] 26

> sum(diff_results$bootstrap_p<=0.05)

[1] 49

> ordfit_adjp <- p.adjust(diff_results$ordfit_pvalue, "BH")
> sum(ordfit_adjp<=0.05)

[1] 0

> perm_adjp <- p.adjust(diff_results$permutation_p, "BH")
> sum(perm_adjp<=0.05)

[1] 0

> boot_adjp <- p.adjust(diff_results$bootstrap_p, "BH")
> sum(boot_adjp<=0.05)

[1] 6

> diff_list_perm <- which(perm_adjp<=0.05)
> diff_list_boot <- which(boot_adjp<=0.05)
> sig_results_perm <- cbind(ovarian_cancer_methylation[diff_list_perm, ], diff_results$ordfit_t[
> print(sig_results_perm)

 [1] IlmnID
 [2] Beta
 [3] exmdata2[, 2]
 [4] exmdata3[, 2]
 [5] exmdata4[, 2]
 [6] exmdata5[, 2]
 [7] exmdata6[, 2]
 [8] exmdata7[, 2]
 [9] exmdata8[, 2]
[10] diff_results$ordfit_t[diff_list_perm]
[11] diff_results$permutation_p[diff_list_perm]
<0 rows> (or 0-length row.names)

> sig_results_boot <- cbind(ovarian_cancer_methylation[diff_list_boot, ], diff_results$ordfit_t[
> print(sig_results_boot)

        IlmnID       Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
146 cg00134539 0.61101320    0.53321780     0.4599934    0.46787420
259 cg00234961 0.04192170    0.04321576     0.0570714    0.05327565
280 cg00260778 0.64319890    0.60488960     0.5673506    0.53150910
```

```
833 cg00814580 0.09348613    0.09619816     0.1201044    0.11534240
887 cg00862290 0.43640520    0.54047160     0.6078680    0.56325950
979 cg00945507 0.13432250    0.23854600     0.3474976    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
146    0.67191510    0.63137380    0.47929610    0.45428300
259    0.04030003    0.03996053    0.05086962    0.05445672
280    0.61920530    0.61925200    0.46753250    0.55632410
833    0.09577040    0.11598850    0.12860890    0.14111200
887    0.50259740    0.40111730    0.56646700    0.54552980
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
146                              5.394750
259                             -4.052697
280                              4.170347
833                             -3.428319
887                             -3.217939
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
146                                       0
259                                       0
280                                       0
833                                       0
887                                       0
979                                       0
```